The CRISIS Wide-Area Security Architecture

Amin Vahdat

March 18, 1998

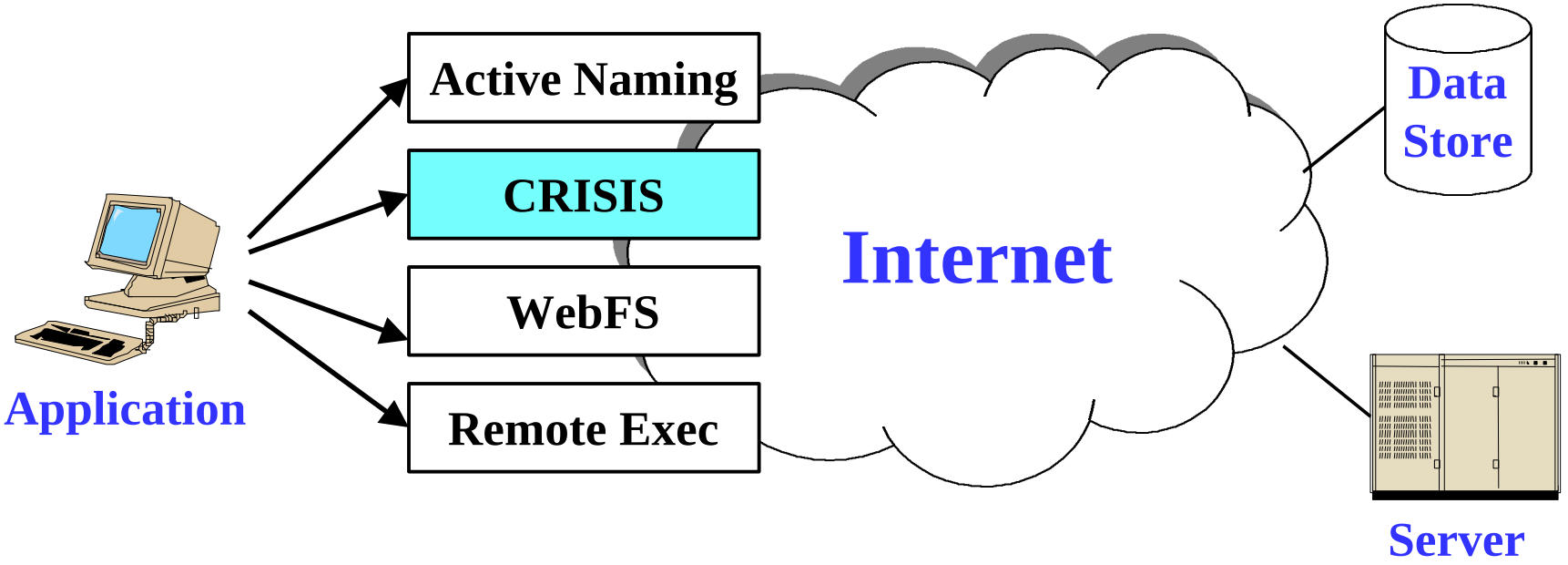
`http://now.cs.berkeley.edu/WebOS`

WebOS: System Support for Wide-Area Applications

Requirement	***WebOS Support***
Migrate data	Coherent persistent storage
Migrate code	Safe remote execution
Locate data/code	Naming
Prevent unauthorized access	Security/authentication

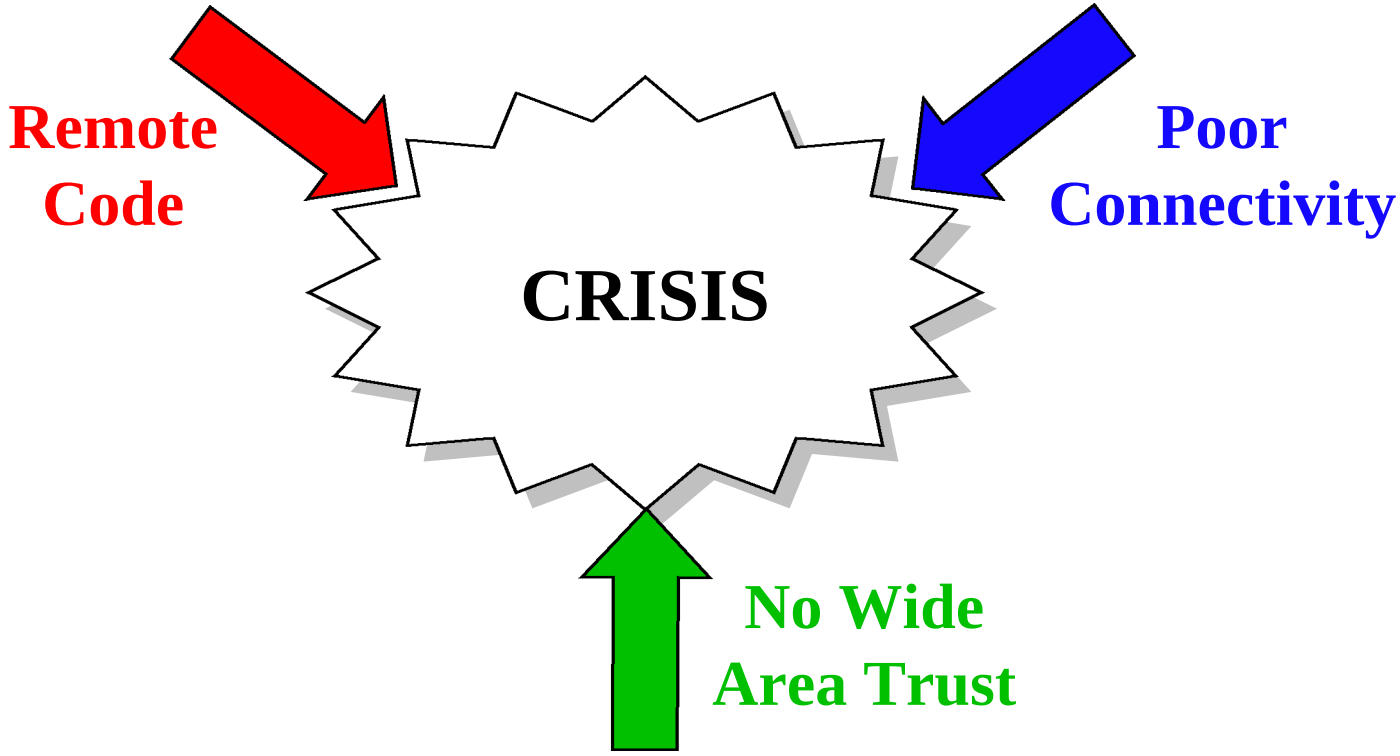
- WebOS provides a common infrastructure for the development and execution of wide-area applications

CRISIS Security



- Secure access to remote programs/data
  - » Rights transfer
  - » Authentication
  - » Revocation

Impending CRISIS for Wide-Area Applications

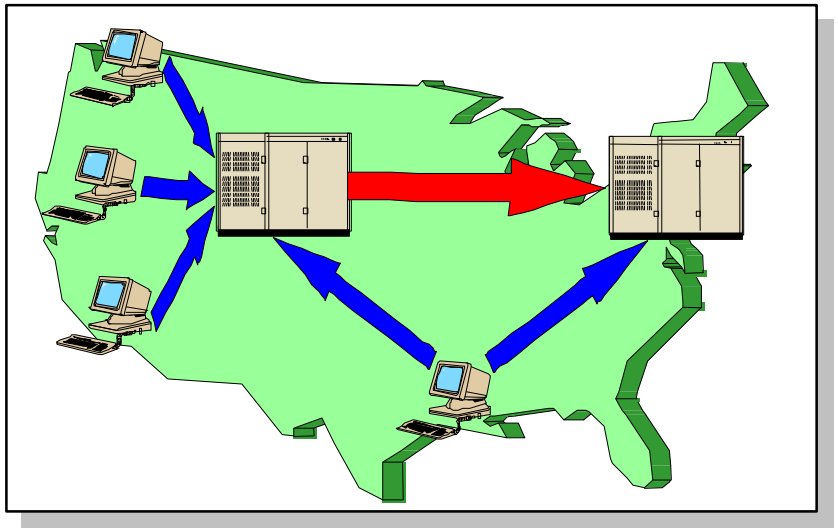


Existing security measures do not match application needs

Outline

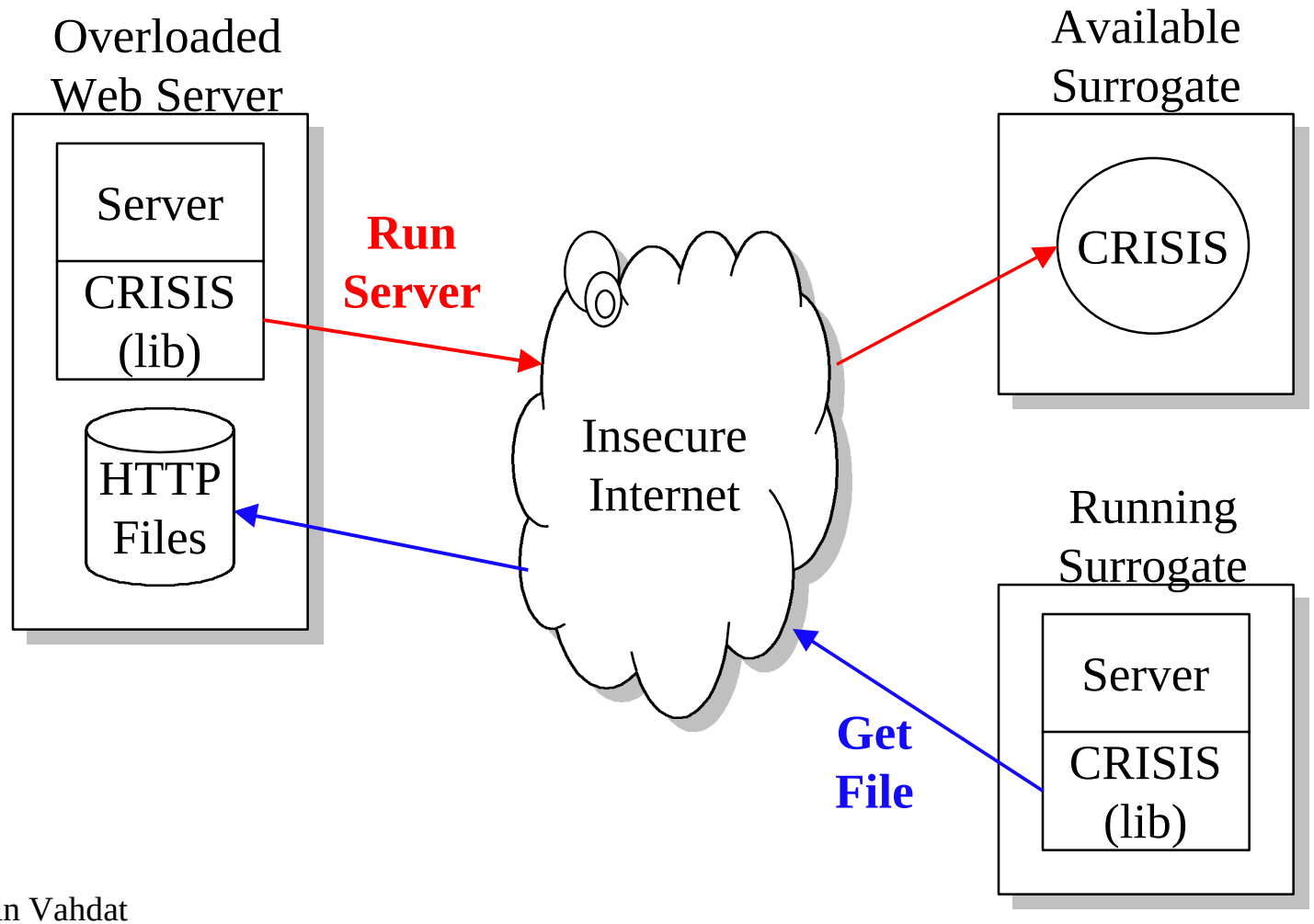
- Problem Statement/Context
- **Motivation (Rent-A-Server)**
- Implementation
- Contributions
- System Scenarios
- Conclusions

Motivating Application: Rent-A-Server



- Dynamically replicate services in response to access patterns
  - » Allocation for peak vs. average utilization
  - » Exploit geographic locality, reduce consumed bandwidth
- Service state securely and automatically distributed
- Transparently choose best replica without user intervention

Rent-A-Server



Rent-A-Server Security Issues

- Remote Process Execution
  - » Buggy/malicious programs
  - » Authorization
- Secure access to sensitive data
  - » Redundancy
- Fine-grained rights transfer
  - » Short-term, revocable rights
  - » Avoid modifying ACL's
- Performance
  - » Avoid validation with central authority on every access

Other Applications

- SchoolNet
- Wide Area Collaboration
- Mobile Login
- Large Scale Remote Execution
- Encrypted Intermediate Caches
- Database access

CRISIS in Context

- Wide-Area characteristics:
  - Availability of remote computation
  - Lack of trust
  - Poor network connectivity
- Wide-area security requirements:
  - Performance/Availability
  - Fine-grained rights transfer
  - Multiple administrative domains
  - Revocation

Outline

- Problem Statement/Context
- Motivation (Rent-A-Server)
- **Implementation**
- Contributions
- System Scenarios
- Conclusions

Alternatives

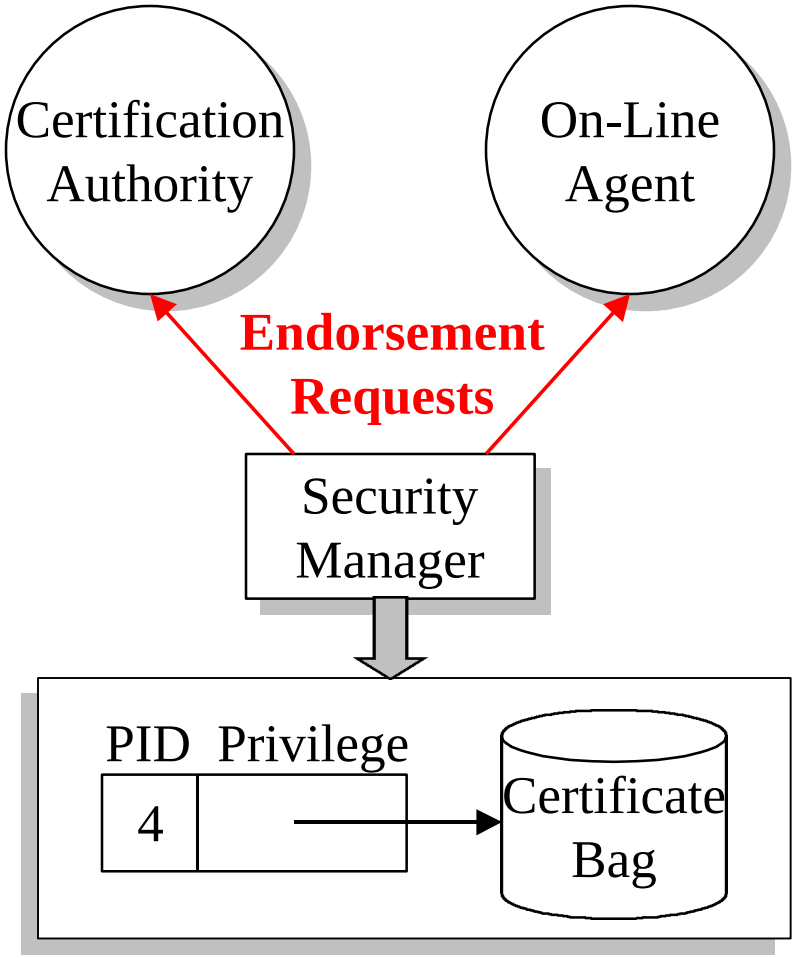
- Secure login
  - » Lacks fine-grained control over access rights
  - » Overhead of creating accounts everywhere
- Kerberos
  - » Synchronous communication with ticket granting servers
  - » Share secrets between administrative domains, single point of attack
- Public Key
  - » How to perform revocation

➡ No reasoning about fine-grained transfer of rights or remotely programmable resources

CRISIS Contributions

- Transfer certificates
  - » Light-weight revocable capabilities
  - » Delegation, roles
  - » Fine-grained rights transfer
- Flexible support for security/performance/availability
- Revocation as first class operation
- Proof-carrying requests
  - » Complete accountability
  - » Simplify authorization

CRISIS Architecture



CRISIS Implementation

- Security managers map processes to *security domains*
  - » Certificate bag describes privileges associated with processes
  - » e.g., new security domain created for a login shell
- Certificates describe privileges
  - » X.509/SSL (public key)
- Dual Endorsement
  - » CA long timeout, offline
  - » OLA short timeout, online
- Simple Revocation

Outline

- Problem Statement/Context
- Motivation (Rent-A-Server)
- Implementation
- **Contributions**
  - **Transfer Certificates**
  - **Authorization**
  - **Revocation**
- System Scenarios
- Conclusions

Problem: Fine-Grained Rights Transfer

- Overloaded Berkeley server harvests Texas surrogate
- Allow Texas to access customer database, not my email
- Avoid modifying ACL's

Transfer Certificates

- Light-weight revocable capabilities
  - » Transfer rights from one principal to another
- Delegation
  - » Databases
  - » sendmail
- Fine-grained rights restriction
  - » Different levels of trust for different nodes
  - » Roles

Access Control Lists vs. Capabilities

- Access Control Lists
  - » Explicitly describe users privileged to access a resource
  - » Issues: Error prone, cumbersome
- Capabilities
  - » Distribute opaque unforgeable ticket granting access
  - » Issues: confinement, revocation
- ➡ Transfer certificates
  - » Explicitly describe privileges transferred (source, destination)
  - » Reference monitor determines if *entire chain* of transfers valid
  - » Revocation as first class operation

Problem: Fine-Grained Rights Transfer

- Overloaded Berkeley server harvests Texas Surrogate
- Avoid modifying ACL's

- Solution:
  - » Berkeley signs xfer certificate stating privileges granted to Texas
- Example:
  - » ACL(inputFile): Berkeley
  - » [Texas may access inputFile]$_{\texttt{Berkeley}}$ [Berkeley Key is x]$_{\text{CA}}$

Problem:
Secure Access to Sensitive Data

- Allow Texas to access customer database, not my email
- Texas must prove it is authorized for DB access

Authorization

- Hybrid ACL/capability approach
  - » ACL's maintain list of authorized principals
  - » Transfer certificates grant revocable capabilities
- Reference Monitor validates chain of transfers
  - » Time
  - » Complete accountability
  - » Path of trust: hierarchical trust
  - » Valid signature (CA)/valid counter-signature (OLA)

Problem:
Secure Access to Sensitive Data

- Allow Texas to access customer database, not my email
- Texas must prove it is authorized for DB access

- Solution, Texas transmits:
  - » Identity certificate: CA says this key speaks for Texas
  - » Transfer certificate: Berkeley says Texas can access DB
- ACL contains only Berkeley entry

Problem: Revoking Rights

- Load subsides, Berkeley discontinues use of Texas
- Berkeley discontinues Texas access to DB
- Protect against future compromise

Revocation

- Valid certificates contain dual signatures
  - » Certification authority: sign with long timeout (offline)
  - » On-Line Agent: sign with short timeout (highly available)
- Certificates cached if both signatures are fresh
  - » Tradeoff security/performance/availability
  - » Redundancy: violate two entities in different ways
- Revocation as first-class operation
  - » Inform On-Line Agent no further endorsement of certificate
  - » Rights revoked modulo timeout of certificate

Problem: Revoking Rights

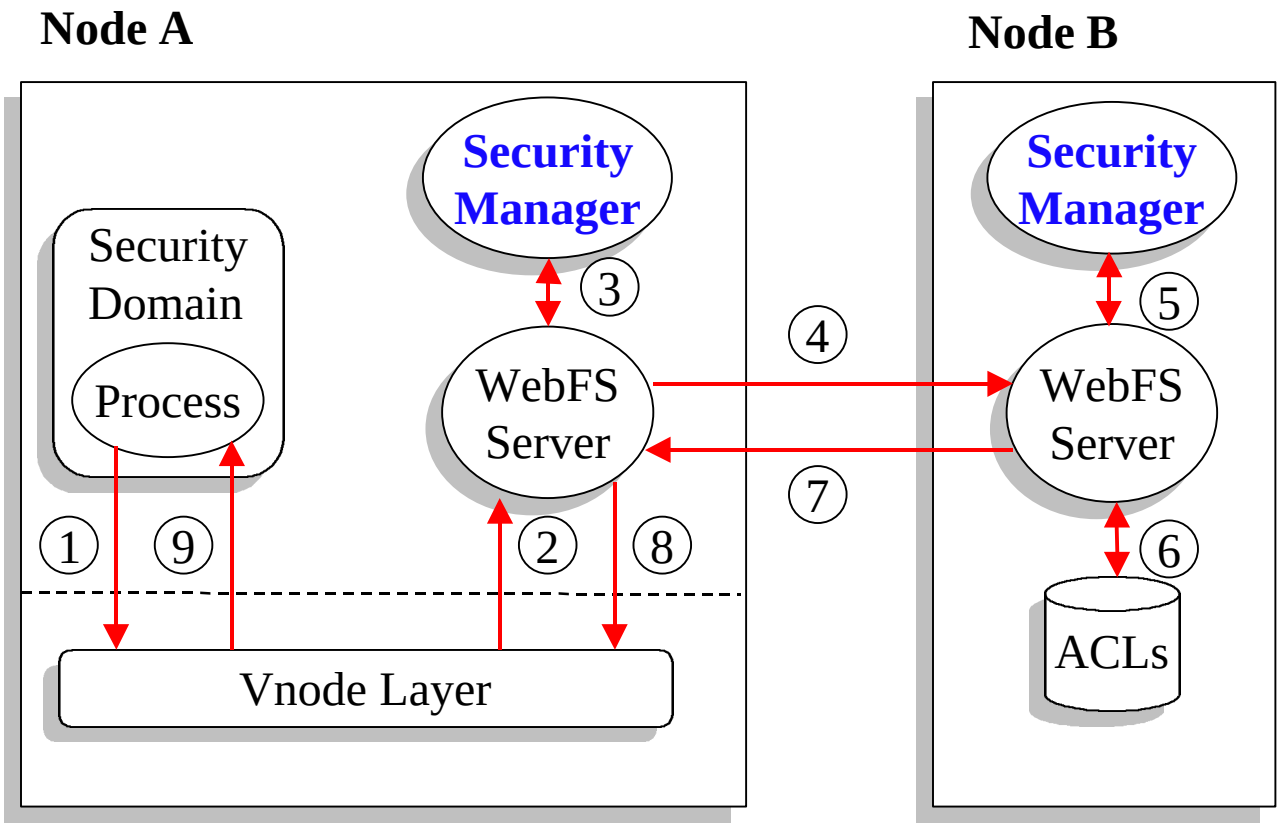
- Load subsides, Berkeley discontinues use of Texas
- Berkeley discontinues Texas access to DB
- Protect against future compromise

- Solution:
  - » Berkeley instructs OLA: no longer endorse transfer certificate

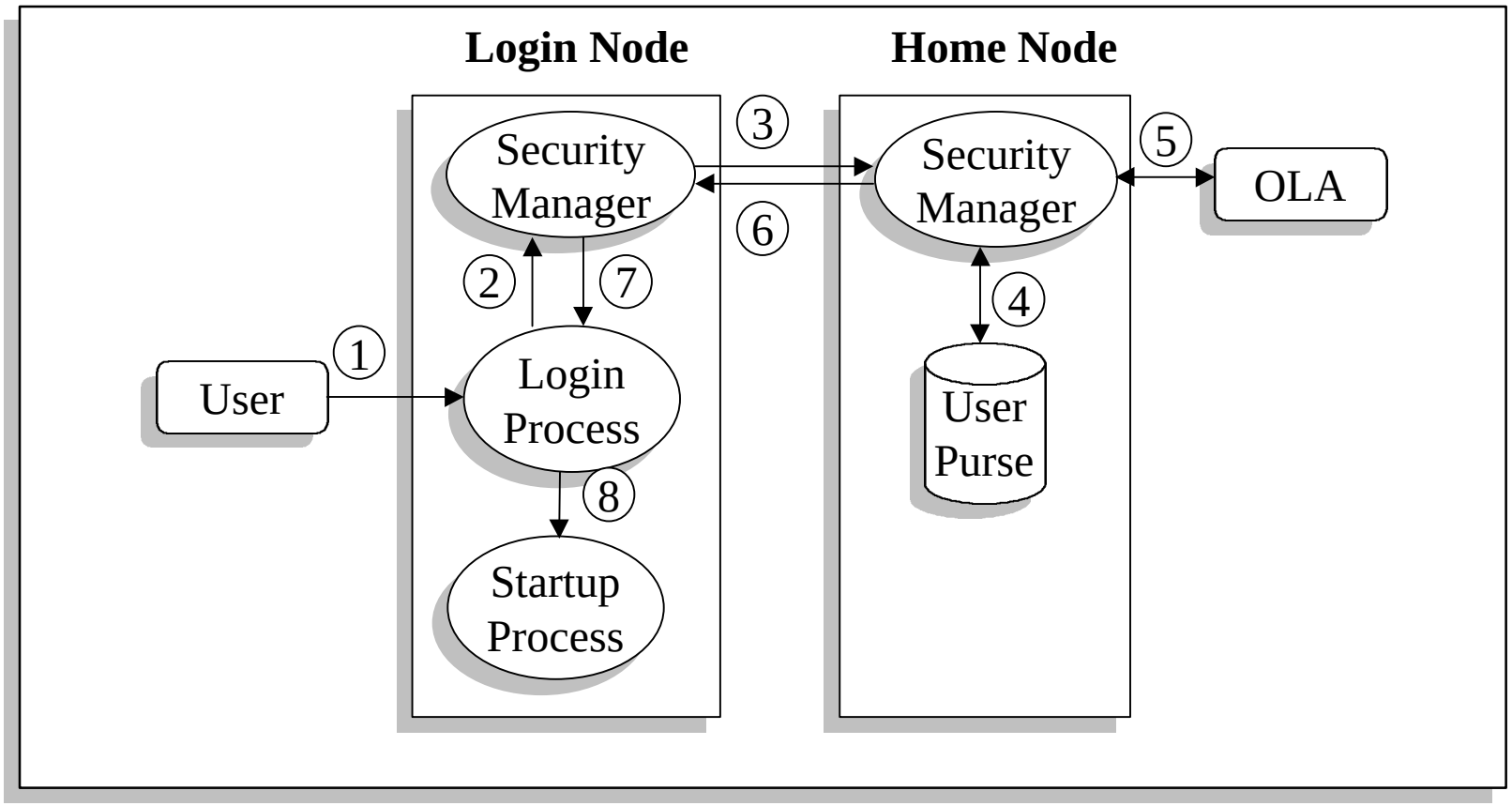
Outline

- Problem Statement/Context
- Motivation (Rent-A-Server)
- Implementation
- Contributions
- **System Scenarios**
  - **File Access**
  - **Login**
  - **Remote Execution**
- Conclusions

Integrating CRISIS with the File System



Login Example



Running Remote Code

- Java: architecture independence
  - » Security classes determine application privileges
  - » Virtual machine rejects disallowed operations
- Janus[Goldberg96]: UNIX compatibility
  - » Solaris **`proc`** filesystem intercepts system calls
  - » User level process disallows “dangerous” system calls
  - » Per-process profile determines dangerous operations
- Future work
  - » Determining least privileges required to complete task
  - » Resource allocation among competing processes

Rent-A-Server: Putting It All Together

- Hierarchical trust among administrative domains
- Secure access to sensitive data (customer DB)
- Execution of programs in sandbox
  - » Protect surrogates from buggy/malicious programs
- Transfer certificates
  - » Fine-grained, short-lived access
  - » No need to modify ACL's
- Proof-carrying requests simplify authorization

Conclusions

- Design and initial implementation of wide-area security system
- Enable secure access to global resources
- Transfer certificates simplify delegation, roles for rights restriction

➡ **`http://now.cs.berkeley.edu/WebOS`**